Major League Baseball Database

Stuart Sandifer – CS 340 Final Project

Outline

My database is a representation of the world of major league baseball with an emphasis on player statistics. Player statistics are recorded for each game that is played. Two teams play a game, which results in a list of offensive statistics for each player that participates, based on each player's performance in different categories. Players are organized into teams and teams into divisions. Players are also categorized by their position. Player statistics that are of interest can be displayed and referred back to as needed. The player data can be organized by team, division, position, and game, and is a valuable reference tool.

Database Outline

The database has six main entities: players, positions, teams, divisions, games, and game statistics. The six relationships include the following: each team has multiple players, each player has one or more positions, each player has player game results, each game has two teams, each game has multiple player game results, and each division has teams.

The game results table, mlb_player_game_result, contains the actual data for each player and has keys for player id and a game id. For every entry in the table, there is a

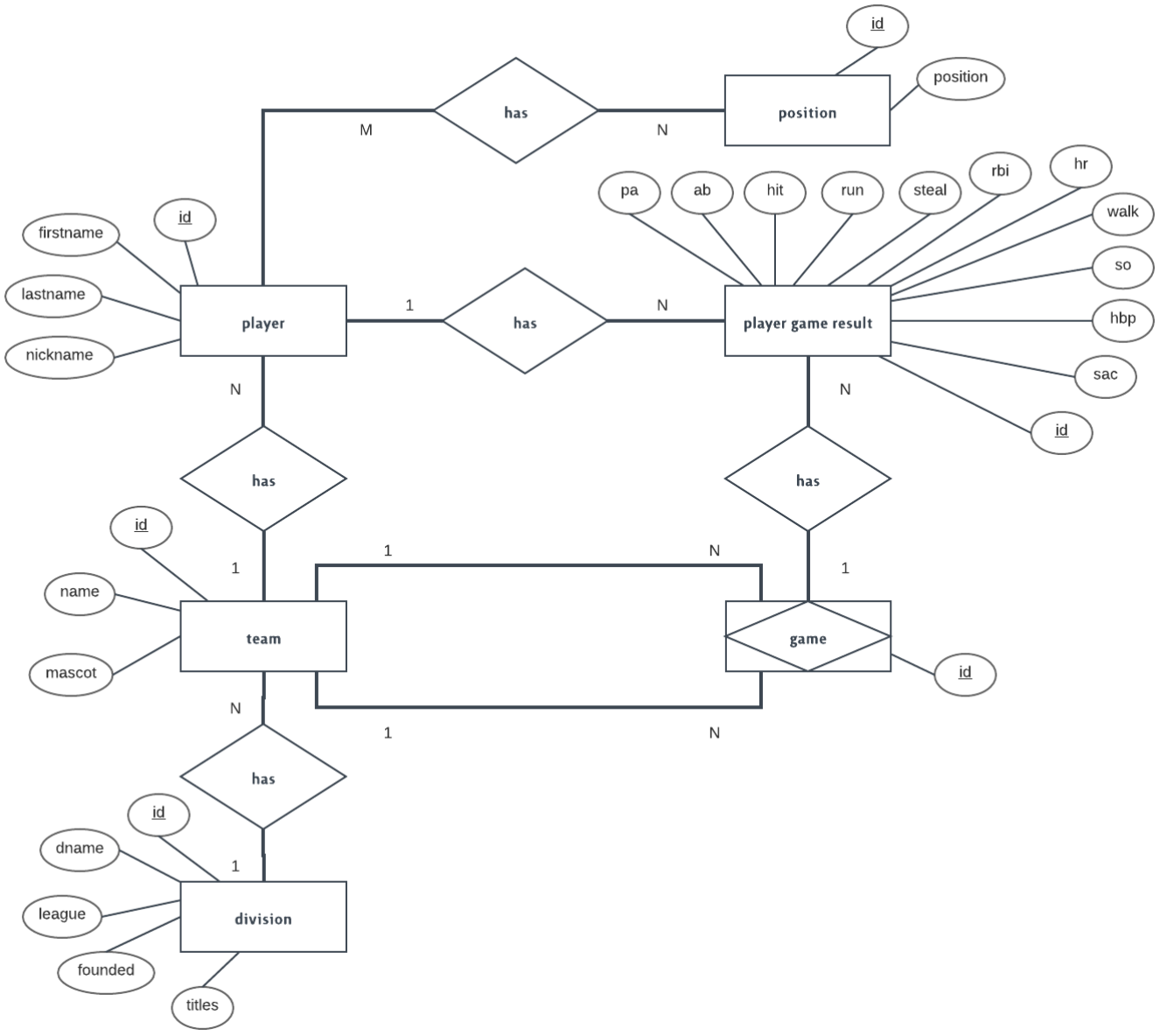
player id and game id associated with those tables. Game statistics are entered from a form where the user must choose a game and a player. The statistics are entered for the selected player for that specific game. Thus the mlb_player_game_result table will have several rows for each player id and several rows for each game id. It is auto-incrementing and has attributes including plate appearances, at bats, hits, runs, steals, runs batted in, home runs, walks, strike outs, hit by pitch, and sacrifice flies. The game id points to the mlb_game table, which keeps track of the two teams that participated in the game. The player id points to the id of the specific player in mlb_player table.

The mlb_player table contains the player's identification: id (auto-incrementing), name, and a key for team id that references the mlb_team table, and the mlb_game table has keys for the two team's ids- from mlb_team- that participated in the game. The mlb_team table contains the team information and a key for the mlb_division id. The mlb_division table contains attributes pertaining to the details about the division and leagues.

There is also a table called mlb_position_player which is a many-to-many relationship between mlb_player and mlb_position. A player can play more than one position and a position has more than one player. The mlb_position_player table creates this relationship using constraints for player id and position id to allow for multiples of both. Each table provides additional levels of sorting options for displaying each aspect of the league and the data stored within.

ER Diagram:

MLB ER Diagram



Schema:


mlb_player_game_result
id int
pa int
ab int
hit int
run int
steal int
rbi int
ht int
walk int
so int
hbp int
sac int
playerid int
gameid int
mlb_position
id int
position string
mlb_position_player
posid int
plid int
mlb_player
id int
firstname string
lastname string
nickname string
team int
mlb_game
id int
teama int
teamb int
mlb_team
id int
name string
mascot string
division int
mlb_division
id int
dname string
league string
founded string
titles string

```
-- ---------------------------------------------
--          MLB Baseball Database
--          Data Definition Queries
-- ---------------------------------------------

CREATE TABLE `mlb_division` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `dname` varchar(255) NOT NULL,
  `league` varchar(255) NOT NULL,
  `founded` int(11) NOT NULL,
  `titles` int(11) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB;


CREATE TABLE `mlb_team` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(255) NOT NULL,
  `mascot` varchar(255) DEFAULT NULL,
  `division` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `name` (`name`),
  CONSTRAINT `mlb_team_fk_1` FOREIGN KEY (`division`) REFERENCES `mlb_division` (`id`) ON DELETE
SET NULL ON UPDATE CASCADE
) ENGINE=InnoDB;



CREATE TABLE `mlb_player` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `firstname` varchar(255) NOT NULL,
  `lastname` varchar(255) DEFAULT NULL,
  `team` int(11) DEFAULT NULL,
  `nickname` varchar(255) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `nickname` (`nickname`)
  CONSTRAINT `mlb_player_fk_1` FOREIGN KEY (`team`) REFERENCES `mlb_team` (`id`) ON DELETE SET
NULL ON UPDATE CASCADE
) ENGINE=InnoDB;


CREATE TABLE `mlb_position` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `position` varchar(255) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB;
```

```
CREATE TABLE `mlb_position_player` (
  `posid` int(11) NOT NULL DEFAULT '0',
  `plid` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`posid`,`plid`),
  KEY `plid` (`plid`),
  CONSTRAINT `mlb_position_player_fk_1` FOREIGN KEY (`posid`) REFERENCES `mlb_position` (`id`),
  CONSTRAINT `mlb_position_player_fk_2` FOREIGN KEY (`plid`) REFERENCES `mlb_player` (`id`)
) ENGINE=InnoDB;
CREATE TABLE `mlb_game` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `teama` int(11) DEFAULT NULL,
  `teamb` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `mlb_game_fk_1` FOREIGN KEY (`teama`) REFERENCES `mlb_team` (`id`),
  CONSTRAINT `mlb_game_fk_2` FOREIGN KEY (`teamb`) REFERENCES `mlb_team` (`id`)
) ENGINE=InnoDB;



CREATE TABLE `mlb_player_game_result` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `pa` int(11) DEFAULT NULL,
  `ab` int(11) DEFAULT NULL,
  `hit` int(11) DEFAULT NULL,
  `run` int(11) DEFAULT NULL,
  `steal` int(11) DEFAULT NULL,
  `rbi` int(11) DEFAULT NULL,
  `hr` int(11) DEFAULT NULL,
  `walk` int(11) DEFAULT NULL,
  `so` int(11) DEFAULT NULL,
  `hbp` int(11) DEFAULT NULL,
  `sac` int(11) DEFAULT NULL,
  `playerid` int(11) DEFAULT NULL,
  `gameid` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `playerid` (`playerid`),
  CONSTRAINT `mlb_player_game_result_fk_1` FOREIGN KEY (`playerid`) REFERENCES `mlb_player` (`id`)
ON DELETE SET NULL ON UPDATE CASCADE,
  CONSTRAINT `mlb_player_game_result_fk_3` FOREIGN KEY (`gameid`) REFERENCES `mlb_game` (`id`)
ON DELETE SET NULL ON UPDATE CASCADE
) ENGINE=InnoDB;
```

```
-- ---------------------------------------------
--          MLB Baseball Database
--          Data Manipulation Queries
-- ---------------------------------------------


-- Used to select a division from a drop-down menu:
SELECT id, league, dname FROM mlb_division ORDER BY league ASC;

-- Used for selecting a team from a drop-down menu:
SELECT id, name, mascot FROM mlb_team ORDER BY name ASC;

-- Used for selecting a position from a drop-down menu:
SELECT id, position FROM mlb_position ORDER BY id ASC;

-- Used for selecting a player from a drop-down menu:
SELECT id, firstname, lastname FROM mlb_player ORDER BY lastname ASC;

-- List player home runs by division based on user input:
SELECT mlb_division.league, mlb_division.dname, mlb_player.firstname, mlb_player.lastname,
SUM(mlb_player_game_result.hr) AS total FROM mlb_player_game_result INNER JOIN
mlb_player ON mlb_player_game_result.playerid = mlb_player.id
INNER JOIN mlb_team ON mlb_player.team = mlb_team.id
INNER JOIN mlb_division ON mlb_team.division = mlb_division.id
WHERE mlb_division.id = [userInput]
GROUP BY mlb_player.id
ORDER BY total desc;

-- List player batting average by team based on user input:
SELECT mlb_team.name, mlb_team.mascot, mlb_player.firstname, mlb_player.lastname,
SUM(mlb_player_game_result.hit)/SUM(mlb_player_game_result.ab) AS avg FROM
mlb_player_game_result
INNER JOIN mlb_player ON mlb_player_game_result.playerid = mlb_player.id
INNER JOIN mlb_team ON mlb_player.team = mlb_team.id
WHERE mlb_team.id = [userInput]
GROUP BY mlb_player.id
ORDER BY avg desc;

-- List player runs batted in by player position based on user input:
SELECT pl.firstname, pl.lastname, pos.position, SUM(gr.rbi) AS total
FROM mlb_position AS pos
LEFT JOIN mlb_position_player AS pospl ON pos.id = pospl.posid
LEFT JOIN mlb_player AS pl ON pospl.plid = pl.id
INNER JOIN mlb_player_game_result AS gr ON gr.playerid = pl.id
WHERE pos.id = [userInput]
GROUP BY pl.id
ORDER BY total desc;
```

```
-- List player batting averages:
SELECT mlb_player.firstname, mlb_player.lastname,
SUM(mlb_player_game_result.hit)/SUM(mlb_player_game_result.ab) AS avg FROM
mlb_player_game_result INNER JOIN
mlb_player ON mlb_player_game_result.playerid = mlb_player.id
GROUP BY mlb_player.lastname ORDER BY avg desc LIMIT 4;

-- Display complete list of players:
SELECT mlb_player.firstname, mlb_player.lastname, mlb_team.name, mlb_team.mascot FROM
mlb_player INNER JOIN mlb_team ON mlb_player.team = mlb_team.id ORDER BY lastname ASC;

-- Add a player and player's positions based on user input:
INSERT INTO mlb_player(firstname, lastname, nickname, team) VALUES ([a],[b],[c],[d]);
INSERT INTO mlb_position_player(plid, posid) VALUES ([e],[f]);

-- Remove and add new player positions based on user input:
DELETE FROM mlb_position_player WHERE plid = [a];
INSERT INTO mlb_position_player(plid, posid) VALUES ([b],[c]);

-- Update a player's current team:
UPDATE mlb_player SET team = [a] WHERE id = [b];

-- Add a team:
INSERT INTO mlb_team(name, mascot, division) VALUES ([a],[b],[c]);

-- Delete a team:
DELETE FROM mlb_team WHERE id = [a];

-- Display all teams by division:
SELECT mlb_division.league, mlb_division.dname, mlb_team.name, mlb_team.mascot FROM mlb_team
INNER JOIN mlb_division ON mlb_team.division = mlb_division.id
ORDER BY league ASC, dname ASC, name ASC;

-- Add a new game based on user input:
INSERT INTO mlb_game(teama, teamb) VALUES ([a],[b]);

-- Add player stats for a game based on user input:
INSERT INTO mlb_player_game_result(playerid, gameid, pa, ab, hit, run, steal, rbi, hr, walk, so, hbp, sac)
VALUES ([a],[b],[c],[d],[e],[f],[g],[h],[i],[j],[k],[l],[m]);

-- Drop-down menu to have user select a game so results can be displayed:
SELECT a.id AS GameID, ta.name AS TeamA_Name, tb.name AS TeamB_Name FROM mlb_game AS a
INNER JOIN mlb_team AS ta ON ta.id = a.teama
INNER JOIN mlb_team AS tb ON tb.id = a.teamb ORDER BY GameID ASC;

-- Display the game results based on user input:
SELECT tm.name, pl.firstname, pl.lastname, gr.hit, gr.run, gr.steal, gr.rbi, gr.hr, gr.walk, gr.so FROM
mlb_game g INNER JOIN mlb_player_game_result gr ON gr.gameid = g.id
```

```
INNER JOIN mlb_player pl ON gr.playerid = pl.id
INNER JOIN mlb_team tm ON pl.team = tm.id
WHERE g.id = [a]
ORDER BY tm.name desc;

-- Display game stats for all players:
SELECT mlb_player.firstname, mlb_player.lastname,
SUM(mlb_player_game_result.hit)/SUM(mlb_player_game_result.ab) AS avg,
SUM(mlb_player_game_result.run) AS runs, SUM(mlb_player_game_result.rbi) AS rbi FROM
mlb_player_game_result INNER JOIN
mlb_player ON mlb_player_game_result.playerid = mlb_player.id
GROUP BY mlb_player.lastname ORDER BY avg desc;
```

Add positions:

```
INSERT INTO mlb_position (position) VALUES ('catcher');
INSERT INTO mlb_position (position) VALUES ('firstbase');
INSERT INTO mlb_position (position) VALUES ('secondbase');
INSERT INTO mlb_position (position) VALUES ('thirdbase');
INSERT INTO mlb_position (position) VALUES ('shortstop');
INSERT INTO mlb_position (position) VALUES ('outfield');
```

Add divisions:

```
INSERT INTO mlb_division(dname,league, founded, titles) VALUES ('East', 'American', 1969, 39);
INSERT INTO mlb_division(dname,league, founded, titles) VALUES ('East', 'National', 1969, 8);
INSERT INTO mlb_division(dname,league, founded, titles) VALUES ('Central', 'American', 1994, 11);
INSERT INTO mlb_division(dname,league, founded, titles) VALUES ('Central', 'National', 1994, 24);
INSERT INTO mlb_division(dname,league, founded, titles) VALUES ('West', 'American', 1969, 14);
INSERT INTO mlb_division(dname,league, founded, titles) VALUES ('West', 'National', 1968, 16);
```

Add teams:

```
INSERT INTO mlb_team(name, mascot, division) VALUES ('New York', 'Yankees', 1);
INSERT INTO mlb_team(name, mascot, division) VALUES ('Baltimore', 'Orioles', 1);
INSERT INTO mlb_team(name, mascot, division) VALUES ('Boston', 'Redsox', 1);
INSERT INTO mlb_team(name, mascot, division) VALUES ('Tampa Bay', 'Rays', 1);
INSERT INTO mlb_team(name, mascot, division) VALUES ('Toronto', 'BlueJays', 1);

INSERT INTO mlb_team(name, mascot, division) VALUES ('Minnesota', 'Twins', 3);
INSERT INTO mlb_team(name, mascot, division) VALUES ('Cleveland', 'Indians', 3);
INSERT INTO mlb_team(name, mascot, division) VALUES ('Detroit', 'Tigers', 3);
INSERT INTO mlb_team(name, mascot, division) VALUES ('Chicago', 'WhiteSox', 3);
INSERT INTO mlb_team(name, mascot, division) VALUES ('Kansas City', 'Royals', 3);

INSERT INTO mlb_team(name, mascot, division) VALUES ('Houston', 'Astros', 5);
INSERT INTO mlb_team(name, mascot, division) VALUES ('Los Angeles', 'Angels', 5);
```

```
INSERT INTO mlb_team(name, mascot, division) VALUES ('Texas', 'Rangers', 5);
INSERT INTO mlb_team(name, mascot, division) VALUES ('Seattle', 'Mariners', 5);
INSERT INTO mlb_team(name, mascot, division) VALUES ('Oakland', 'Athletics', 5);

INSERT INTO mlb_team(name, mascot, division) VALUES ('Washington', 'Nationals', 2);
INSERT INTO mlb_team(name, mascot, division) VALUES ('Atlanta', 'Braves', 2);
INSERT INTO mlb_team(name, mascot, division) VALUES ('NY', 'Mets', 2);
INSERT INTO mlb_team(name, mascot, division) VALUES ('Miami', 'Marlins', 2);
INSERT INTO mlb_team(name, mascot, division) VALUES ('Philadelphia', 'Phillies', 2);

INSERT INTO mlb_team(name, mascot, division) VALUES ('Chi', 'Cubs', 4);
INSERT INTO mlb_team(name, mascot, division) VALUES ('Milwaukee', 'Brewers', 4);
INSERT INTO mlb_team(name, mascot, division) VALUES ('St Louis', 'Cardinals', 4);
INSERT INTO mlb_team(name, mascot, division) VALUES ('Cincinnati', 'Reds', 4);
INSERT INTO mlb_team(name, mascot, division) VALUES ('Pittsburgh', 'Pirates', 4);

INSERT INTO mlb_team(name, mascot, division) VALUES ('Colorado', 'Rockies', 6);
INSERT INTO mlb_team(name, mascot, division) VALUES ('LA', 'Dodgers', 6);
INSERT INTO mlb_team(name, mascot, division) VALUES ('Arizona', 'Diamonbacks', 6);
INSERT INTO mlb_team(name, mascot, division) VALUES ('San Francisco', 'Giants', 6);
INSERT INTO mlb_team(name, mascot, division) VALUES ('San Diego', 'Padres', 6);
```